REMARKS

The following Response After Final Rejection is submitted in reply to the Final Office Action of January 18, 2005 in the continued examination of the above identified Application and within two months of the mailing of the Final Office Action.

First, in the previous Office Action the Examiner has rejected claim 11 under 35 U.S.C. 112 and the Applicant amended claim 11 accordingly. The Applicant thanks the Examiner for reconsidering and withdrawing the rejection of amended claim 11 under 35 U.S.C. 112.

Claims 1-16 continue to be pending in the present Application and the Examiner has sustained the previous rejection of claims 1, 3, 5, 9, 11, 13 and 15 under 35 U.S.C. § 102 over U.S. Patent No. 6,205,449 to Rastogi et al. for a SYSTEM AND METHOD FOR PROVIDING HOT SPARE REDUNDANCY AND RECOVERY FOR A VERY LARGE DATABASE MANAGEMENT SYSTEM, hereafter referred to as "Rastogi et al. '449", declaring the Applicant's previously submitted arguments distinguishing the present invention over Rastogi et al. '449 as not being persuasive.

The Examiner has also sustained the previous rejection of claims 2, 4, 6, 8, 10, 12, 14 and 16 under 35 U.S.C. § 103(a) over Rastogi et al. '449 and further in view of U.S. Patent No. 5,513,314 to Kandasamy et al. for a FAULT TOLERANT NFS SERVER SYSTEM AND MIRRORING PROTOCOL, hereafter referred to as "Kandasamy et al. '314", declaring the Applicant's previously submitted arguments distinguishing the present invention over Kandasamy et al. '314 as not being persuasive.

The Examiner has also kindly provided discussions in support of the Examiner's finding that the Applicant's previously submitted arguments regarding Rastogi et al. '449 and Kandasamy et al. '314 are not persuasive, for which the Applicant thanks the Examiner.

The Applicant acknowledges and respectfully traverses the raised rejections in view of the following remarks.

First, the Applicant notes and acknowledges that the Examiner has requested that the Applicant not repeat the previously submitted arguments in rebuttal of the Examiner's grounds

for rejection of the claims on the grounds that the repetition of previously submitted arguments is repetitive and does not advance prosecution of the present Application.

In response, and In order to advance prosecution of the present Application by reducing the volume of material the Examiner must consider in the present Response to the present Final Action, and to thereby focus the presently presented arguments and amendments on the specific matters newly raised by the Examiner, the Applicant will not herein repeat the text of the arguments presented in the previous Response.

The Applicant also notes, however, that in paragraphs 3 through 5 of the present Office Action the Examiner has repeated the same grounds for rejection under 35 U.S.C. 102 and 35 U.S.C. 103 that the Examiner stated in previous Office Actions, so that the stated grounds for rejection are still in force at this time.

As will be addressed in the following discussions, it is the belief and position of the Applicant that the previously presented arguments do in fact effectively present a full and adequate rebuttal of the previously stated and presently repeated grounds for rejection under 35 U.S.C. 102 and 103 and fully express the patentable distinctions of the present invention as recited in the claims over the cited prior art. The Applicant therefore expressly does not discard, abandon or concede the arguments raised and presented by the Applicant in the preceding Responses, and therefore respectfully requests that the previously submitted arguments be incorporated herein by reference that the previously considered arguments therefore be considered a part of the present Response.

The Applicant further wishes to point out that while the arguments presented in the previous Response were based upon previously submitted arguments, the arguments submitted in the previous Response were not merely repetitive to previously submitted arguments. As discussed in the previous Response, the Applicant's arguments as submitted therein were in fact amended and expanded over previously presented versions of the arguments in an attempt to clarify the arguments and make the distinctions over the prior art more explicit to the Examiner, as was stated in the previous Response. The Applicant assumes that the Examiner

has noted and considered the extension and clarification of the arguments that was presented in the previous Response.

The following will therefore primarily address the Examiner's specific comments and remarks concerning the previous Response as the Examiner's comments and remarks contained in paragraphs 7 through 26 of pages 5 through 16 of the present Office Action appear to comprise the actual new issues raised by the Examiner. The following will thereby refer to the Applicant's previously submitted arguments only as necessary to address the Examiner's comments and remarks.

Paragraph 7:

Referring therefore to paragraph 7 of the present Final Action, the Examiner has disagreed with the Applicant's characterization of state machines in the presently pending claims and cites the textbook Logic and Computer Design Fundamentals by Mano and Kime for the definition that a state machine consists of a sequence of states rather than as a sequence of state machines.

In this regard, an invention is an advance or development over the prior art and can typically embody new concepts or viewpoints not anticipated of perceived in the prior art. As such, a prior art reference, such as a textbook, can by its very nature describe and define only what is in the prior art and an invention cannot be defined and bound only to statements in textbooks or there would be few or no new inventions. In addition, and referring to the above discussions of state machines in the context of the present invention, the specification of the present Application defines the terms "state" and "state machine" according to the present invention and the Applicant is entitled to recognition of terms so defined so long as those definitions do not explicitly mis-represent the terms as understood in the art. As discussed above, the terms "state" and "state machine" as define and employed in the present Application are not in conflict with those terms as understood in the art and are instead developments and extensions of the meanings of those terms over what was known in the prior art.

More specifically, the present invention defines a state machine, for purposes of the present invention, as representing the state existing in a system during one step in the execution of an operation comprised of a sequence of such steps. As discussed, this definition is not in conflict with the classical definition of a state machine as stated in, for example, Logic and Computer Design Fundamentals, by Mano and Kime, but represents a new concept of the relationship between state machines and operations or processes executed in a system.

In this regard, the Applicant concurs that a state machine is classically defined by the outputs resulting from each possible combination of inputs, and that the set of all outputs for all possible inputs represents the internal logic of the state machine. The definition of a state machine according to the present invention is not, however, in conflict with the classic definition of a state machine as consisting of a sequence of states as found, for example, in Logic and Computer Design Fundamentals, by Mano and Kime and as summarized by the Examiner in paragraph 7 of the Final Action.

The present invention, however, has recognized that a "sequence of states" defining a state machine may be comprised of a single state, that is, of one specific combination of inputs and the corresponding one specific combination of outputs. A state machine of the present invention as defined in the context, specification and claims of the present invention is accordingly defined for the purposes of the present invention as a single step state machine wherein a given single step state machine is, in agreement with the definition of a state machine, defined by the state existing during a single step of a sequence of one or more steps that are executed in the execution of, for example, a file transaction operation.

The meaning of the terms "state" and "state machine" as defined and used in the specification and claims of the present invention are therefore fully defined in the specification and claims of the present Application and are not in conflict with the classical definitions of these terms, but are instead a new extension and development from the terms as classically employed.

Returning to the issue originally expressed by the Examiner, the Applicant notes that the Examiner apparently understands the Applicant as defining a state machine as a sequence of state machines, which the Examiner believes to be incorrect.

In response, it must first be noted that at no place in the claims does the Applicant recite a state machine as a sequence of state machines. The claims instead recite that an operation, such as a transaction operation, is comprised of a sequence of one or more steps and that the system of the present invention represents each transaction operation as a sequence of state machines wherein each state machine represents and is defined by the state in a single step in the sequence of steps executed in executing a transaction operation. This is very different from reciting that a state machine is a sequence of state machines.

The Applicant therefore cannot see how the Examiner's understanding of a state machine as a sequence of state machines was derived from the claims of the present invention. The Applicant must therefore assume that the Examiner arrived at this misunderstanding from the Applicant's arguments that were presented to explain the distinctions between the present invention and the prior art. If so, the Applicant must apologize for confusing the Examiner and hopes that the above discussions clarify what is meant by "state" and "state machine".

The Applicant must point out, however, that the statement that a state machine is comprised of a sequence of state machines is correct according to the present invention, and is in fact one of the principle concepts of the present invention by which the present invention is distinguished over the prior art as taught, for example, in Mano and Kime.

This is the point that the Applicant has attempted to explain in previous Responses and the Applicant apologizes if this point has not been made sufficiently clear. The following is therefore an attempt to further clarify the point that, according to the present invention a state machine may be comprised of a sequence of state machines.

Therefore continuing with above discussion of the concept of state and state machines according to the present invention, if a system is defined as a state machine, then according to Mano and Kine the system state machine is defined by all possible states that could exist in

the system, which in turn is represented by all possible combinations of system inputs and the corresponding system outputs. It will be apparent that all combinations of system inputs and the corresponding system outputs that can exist in the system will include all operations that can be performed by the system wherein each operation will be comprised of a sequence of states selected from all of the states that can exist in the system state machine. According to the present invention, therefore, each operation that can be executed by the system state machine may be regarded as itself comprising an operation state machine wherein an operation state machine is defined by the sequence of system states that exist during execution of the operation and wherein the operation states and their sequence are selected from the set of all states that can exist in the system state machine.

Continuing with this concept, and according to the present invention, it will also be apparent that each operation that can be performed by the system state machine will be comprised of a sequence of steps that are executed by the system state machine in performing the operation. Each step, however, is represented by the state existing in the system during the execution of that step and, according to the present invention, is therefore represented by a state machine that is defined by the state existing during execution of that step, which may be referred to as a "step state machine". An operation performed by a system, that is, by a system state machine, is thereby represented by a sequence of "step state machines" wherein each step state machine represents the state existing in the system state machine during execution of the corresponding step of the operation. According to the present invention, therefore, an operation, which can be represented by an operation state machine, will be comprised of a sequence of steps wherein each step is defined by the state existing during execution of that step and wherein the state existing during each step defines a step state machine representing the step. The operation state machine is therefore, and according to the present invention, comprised of a sequence of step state machines.

The Applicant hopes that the above discussions have clarified and adequately addressed the issue raised by the Examiner in paragraph 7 of the present Final Action and, if the Examiner has any further questions, the Examiner is invited to contact the undersigned.

Paragraph 8:

Therefore next considering paragraph 8 of the present Action, the Examiner does not agree with the Applicant's position and statements that Rastogi et al. '449 does not record or restore the intermediate sub-operations within a transaction. In support of this issue, the Examiner refers to column 8, lines 24-36 of Rastogi et al. '449 wherein Rastogi et al. '449 refers to a recovery algorithm that maintains a separate undo log and redo log in main memory for each transaction.

In response, the Applicant would like to first point out that the existence or non-existence of undo and redo logs says nothing in itself about what information is stored in the logs. In that respect, therefore, the portions of Rastogi et al. '449 cited by the Examiner in this paragraph actually have no information pertinent to the issue of whether or not Rastogi et al. '449 records or restores intermediate sub-operations within a transaction. The existence of undo and redo logs merely means that Rastogi et al. '449 stores information to redo or undo a transaction at some level of detail, but not what that level of detail is or what the stored information is.

The significance of the undo and redo logs in the Rastogi et al. '449 must therefore be determined, if possible, from other portions of the Rastogi et al. '449 specification than just lines 24-36 of column 8. In this regard, however, and as will become apparent from the Rastogi et al. '449 specification and the following discussions, Rastogi et al. '449 essentially does not contain sufficient description of either the undo or redo logs or the information stored therein to determine the nature of the information stored therein, so that any conclusions about the redo and undo logs and the type of information logged in the Rastogi et al. '449 system must be deduced from the Rastogi et al. '449 specification as a whole.

For example, it must be noted that Rastogi et al. '449 does not even mention "state" or "state machine" in the meaning used in the present invention at any place in the specification,

claims or drawings, so that Rastogi et al. '449 is clearly not storing system state or state machines in any form as Rastogi et al. '449 does not even refer to the concepts of operation state or state machines. More specifically, the only mention that Rastogi et al. '449 makes of the term "state" is when Rastogi et al. '449 is at column column 3, lines 22-35, and at column 5, lines 9-65, wherein Rastogi et al. '449 defines the term "state" as bits of information stored in the primary and secondary systems that indicate, in each system, whether the system is the current primary system or is the current secondary system and whether the two computer systems are in "synchronization", that is, have matching copies of the primary computer system transaction log. That is, in the Rastogi et al. '449 system and at any given time only one of the two computer systems can be the current primary computer system, that is, the computer system that is actually executing system transactions while the other system is idle as a spare backup system with no function other than to store the committed "log" of completed transactions.

It will, therefore, be apparent to those of ordinary skill in the arts that in the teachings of Rastogi et al. '449 the term "state" has no relationship or meaning with regard to the state of operation of a system at each step in executing a transaction, or even to the actual execution of transactions, but instead relates only to the overall current functional assignments of the two systems and, in particular, to which one is assigned to execute transactions and which one is assigned to store a copy of the transaction log. "State" in Rastogi et al. '449 therefore has nothing to do with the actual execution of operations or transactions by the primary system, but only with respect to which system is executing the transactions.

Further in this regard, the fact that Rastogi et al. '449 employs the term "state" and explicitly describes the meaning of the word "state" in the specification of the Rastogi et al. '449 patent indicates very clearly that Rastogi et al. '449 did not even consider any other meanings of the term "state" other than that explicitly described in Rastogi et al. '449. In fact, the lack of any definition of the term "state" other than in the sense of identifying which system was primary and which system was secondary indicates quite clearly that Rastogi et al. '449 was not even

aware in any way of the concept or meaning of "state" or "state machine" as employed in the present Application. This clearly further shows that that Rastogi et al. '449 did not and could not intend to describe the logging of "state" or the use of "state machines" as employed in the present invention.

Considering other parts of the Rastogi et al. '449 specification for enlightenment concerning the significance of the undo and redo logs, column 7, line 49 through column 8, line 2, for example, describes that the Rastogi et al. '449 system does have a redo log and an undo log, and states that the system further includes a "dirty page table" that maintains a record of what memory pages have been updated since the last checkpoint, that is, the last time the memory and cache pages were brought up to date together. The description indicates, but does not describe in detail, that the operations of the "dirty page table" are associated in some way with the operations of the undo and redo logs, that is, that the memory and cache management functions are somehow intertwined with the undo and redo logs.

At column 8, lines 3-15, Rastogi et al. '449 defines a "transaction" as being comprised of a sequence of operations at some level in the system and states that the Rastogi et al. '449 system records "transactions", but does not state whether the system records transactions at the "Lo" level, that is, at the level of the transaction itself, as an entity, or at some lower "Lz" level consisting of the sub-operations within a transaction. In this regard, it should be noted that a "transaction" may be undone or redone at any level, ranging from the level of the transaction itself to some level of sub-operation within the transaction, and that the information stored in the undo or redo log will dependent upon the level at which the transaction is redone or undone. At no point does Rastogi et al. '449 explicitly state the level of the transactions at which the redo and undo logs operate, so that this issue must be resolved, in so far as possible, from other aspects of the undo and redo log operations.

In this regard, Rastogi et al. '449 states in lines 15 through 68 of column 8 that transactions are handled and logged depending on whether the transactions are "precommit" or "commit" transactions. That is, in Rastogi et al. '449 a transaction is "precommit" when it has

been stored in the system log in memory and is "commit" when it has been transferred from memory to the stable log, which is effectively on the hard disk of the primary system, where it may still be vulnerable. In this regard, the Rastogi et al. '449 the primary system transmits a copy of the primary system transaction records, that is, a log, to the secondary system through a network connection each time the transaction log in the primary system is "flushed" from the primary system memory space to the primary system mass storage device and when the primary system has transmitted a copy of the transaction log to the secondary system.

It is also described in lines 15 through 68 of column 8 that the operation of the undo and redo logs is intertwined with the cache/memory page update mechanisms in at least that updates to the cache and memory pages are recorded in the undo and redo logs. While the inter-operation of the cache/memory page update mechanisms and the undo/redo logs say nothing directly about the level of information the logs store, the discussion thereof by Rastogi et al. '449 provides some indication of the level of operation of the logs.

That is, and first considering the redo log and recovery mechanism, Rastogi et al. '449 states that a transaction "precommits", that is, is stored in the log in primary system memory when the first sub-operation of the transaction "pre-commits", that is, enters primary system memory to be executed. This statement indicates that the information that is committed to the log includes only the information available at the start of the operation and thereby cannot not include information occurring during the execution of the operation, such as sub-operation information, as that information has not even been brought into primary system memory, much less executed, at the time the transaction is committed to the log. This indicates that the information stored in the log does not include information generated during the execution of the transaction, that is, information from the execution of the sub-operations.

In this regard, it must be noted that the minimum information that is required to implement a redo mechanism is the initial command or instruction determining the transaction to be performed and the initial data used in the transaction, both of which are available when the transaction is initiated and before the first sub-operation is executed and which are sufficient

allow a transaction to be redone from the start. The level of log information is therefore in agreement with the described operation of the pre-commit mechanisms, thereby indicating that the Rastogi et al. '449 system most probably does not store state or any other form of information derived from the execution of sub-operations within a transaction.

This conclusion is further supported when it is noted that while Rastogi et al. '449 does refer to the sub-operations of an operation or transaction when referring to the undo log, such as at column 8, lines 37-41, which indicates that the undo mechanism operates at the sub-operation level, at no point does Rastogi et al. '449 refer to the sub-operations of an operation or transaction when describing the redo log.

Next consider the undo log, it is possible that the undo log may store information regarding current operations of the system in greater detail than does the redo log. Rastogi et al. '449 states, for example, at column 10, lines 8-13, that when a transaction aborts the updates/operations described by the undo log are undone by traversing the undo log sequentially from the end and by executing, in reverse order, every undo record as if the execution were part of the transaction. Rastogi et al. '449 also refers to the sub-operations of an operation or transaction when referring to the undo log, such as at column 8, lines 37-41, which indicates that the undo mechanism operates at the sub-operation level.

It must be noted, however, that the undo log and recovery mechanism in the Rastogi et al. '449 system operate in a completely different manner than does the redo log and recovery mechanism and that the undo log is, in fact, not even a functional part of the operation logging/mirroring mechanism in the sense of a logging/mirror mechanism as described by the present invention or even in the sense described by Rastogi et al. '449. That is, Rastogi et al. '449 specifically states, for example, at column 8, lines 24-44, that the undo log for a transaction is deleted when the transaction "pre-commits". In other words, the undo log is not a part of a long term logging mechanism wherein the logs of transactions are stored for at least an extended period after the transaction is completed. The undo log and recovery mechanism is instead merely a short term, temporary buffer storage in memory that exists only during and

before the execution of the transaction, which is typically adequate for the purposes of an undo log.

In summary, therefore, and for the reasons discussed above, it is the belief and position of the Applicant that Rastogi et al. '449 does not provide sufficient description of either the redo log or the undo log to comprise a teaching with regard to the present invention. It is further the belief and position of the Applicant that in so far as Rastogi et al. '449 contains a description of a redo log and mechanism or an undo log and mechanism that description supports the conclusion and position of the Applicant that the teachings of Rastogi et al. '449 pertinent to the Rastogi et al. '449 redo log are not pertinent to and do not disclose or even suggest the present invention under either 35 U.S.C. 102 or 35 U.S.C. 103.

In further summary, and again for the reasons discussed above, it is the belief and position of the Applicant that the teachings of Rastogi et al. '449 pertaining to an undo log are not pertinent to the present invention because the Rastogi et al. '449 undo log and recovery mechanism perform an entirely different function than does the log mechanism of the present invention.

It must also be noted that the present invention is further distinguished over and from the teachings of Rastogi et al. '449 because the present invention employs a single mechanism to support both the "redo" and "undo" functions. In fundamental contrast from the present invention, and as discussed in detail above, the Rastogi et al. '449 system requires two separate mechanisms to support these functions.

Paragraph 9:

Referring now to paragraph 9 of the present Final Action, the Examiner notes that although the Applicant has used the term "sub-operation" in the arguments previously presented in discussing the distinctions of the present invention over Rastogi et al. '449, that express term does not appear in the claims themselves.

In response, the wishes to point out that the Applicant used the terms "sub-operation" in the discussions and arguments distinguishing the present invention over the cited prior art

in an attempt to clarify the subject matter to the Examiner, believing that the term "sub-operation" might be more familiar than the term "step", as in "step in the execution of an operation". As the Examiner will be aware from reviewing the specification of the present application, "step" is the term used in the specification and claims to refer to a sub-operation executed during and as part of the execution of an operation, such as a file transaction.

The Applicant apologizes for confusing the Examiner, and would appreciate if the Examiner would read the term "step" in replacement of the term "sub-operation" as it appears in the arguments discussing the distinctions of the present invention over the cited prior art, such as in "step in the execution of an operation".

It should be noted that the Applicant would prefer to retain the use of the term "step" to refer to a sub-operation in the execution of an operation as the term "step" is consistent with the terminology used to described the invention in the specification of the present invention. If the Examiner would prefer, however, the Applicant would be willing to amend the language of the claims to use the term "sub-operation" together with the explanatory statement "wherein a step is a sub-operation executed in the execution of an operation".

In conclusion, therefore, the limitations recited in the claims employ the term "step" to refer to a "sub-operation" executed in the execution of an operation and does so in order to conform to the terminology of the specification. It is therefore the belief and position of the Applicant that a limitation referring to a sub-operation in the execution of an operation does in fact appear in the claims.

Paragraph 10:

In paragraph 10 the Examiner objects to the Applicant's arguments distinguishing the present invention over the cited prior art and, in particular, with respect to the Applicant's statement that "in fundamental contrast from the typical from of "back-up" systems as taught by, for example, Rastogi et al. '449, the full processing power of the two control/processing sub-systems is available at all times to process requests from clients. If one control/processing sub-system fails, the other control/processing sub-system continues operative to perform the

request directed to it and to maintain the state machine information necessary to subsequently restore the failed control-processing sub-system." The Applicant stated therein that the system of the present invention was thereby distinguished over and from Rastogi et al. '449 because in the Rastogi et al. '449 system only one of the two systems comprising the total system was operative to process requests as the other of the two systems was essentially idle except for functioning as a log backup to the system processing requests.

The Examiner states that this argument is irrelevant since "this limitation doesn't appear anywhere in the claims".

In response, the Applicant would like to point out that this limitation expressly appears in claims 3 and 7 in the recitation of

"first and second control/processing sub-systems operating concurrently and in parallel, each including a file system processor performing file transaction operations in response to client requests directed to the first and second control/processing sub-systems and controlling file storage operations of the storage sub-system," and

in claims 11 and 15 in the recitation of

"the system resource including a system resource sub-system and first and second control/processing sub-systems, each including a system processor performing system resource operations in response to client requests directed to the first and second control/processing sub-systems and controlling operations of the system resource sub-system".

The Applicant therefore respectfully disagrees with the Examiner as this argument is therefore relevant to at least independent claims 3, 7, 11 and 15. Also, the Applicant believes that there is no requirement that a given argument distinguishing an invention over the prior art must apply to every claim under consideration in order to be a relevant argument.

Paragraph 11:

In paragraph 11 the Examiner refers to the description by Rastogi et al. '449 at column 10, lines 52-61 regarding the roll-back of active transactions. The Examiner maintains that this means that all completed operations that have been directly invoked by the transaction or that

have been directly invoked by an incomplete operation have to be rolled back, and that this means that the current state of a transaction is maintained.

In reply, the Applicant refers the Examiner to the above discussion of paragraphs 7 and 8 and of the undo log and recovery mechanism.

Further in this regard, it should be noted that the operation of the undo mechanism as discussed herein above and as described by Rastogi et al. '449 does not conflict with the Examiner's conclusion that all completed operations that have been directly invoked by a transaction or that have been directly invoked by an incomplete operation have to be rolled back or that the current state of a transaction is maintained in order to do so.

As discussed herein above, however, the undo log and recovery mechanism in the Rastogi et al. '449 system operate in a completely different manner than does the redo log and recovery mechanism and that the undo log is, in fact, not even a functional part of the operation logging/mirroring mechanism in the sense of a logging/mirror mechanism as described by the present invention or even in the sense described by Rastogi et al. '449. That is, Rastogi et al. '449 specifically states, for example, at column 8, lines 24-44, that the undo log for a transaction is deleted when the transaction "pre-commits". In other words, the undo log is not a part of a long term logging mechanism wherein the logs of transactions are stored for at least an extended period after the transaction is completed. The undo log and recovery mechanism is instead merely a short term, temporary buffer storage in memory that exists only during and before the execution of the transaction, which is typically adequate for the purposes of an undo log.

Considering the operation of the undo log and recovery mechanism in terms of operations that have been invoked during the execution of a completed or even uncompleted operation, it is apparent that such invoked operations are effectively part of the invoking operation and, as such, are logged on the undo log together with the invoking operation. Again, however, the undo log contains only the currently active operation, and any operations that are part of, that is, invoked by, the currently active operation. As such, any operations invoked by

a completed but currently active operation or an uncompleted but currently active operation will be discarded with the active operation when the active operation is concluded in one way or another.

Again, therefore, and as stated herein above, it is the belief and position of the Applicant that the structure and operation of the undo log and recovery mechanism is irrelevant to the present invention as the undo log and mechanism are not part of a logging and mirror mechanism in the sense of either the present invention or even the redo log and mechanism of Rastogi et al. '449, but are an entirely different mechanism that may work in conjunction with a redo log and mechanism for certain specific purposes.

Paragraph 12:

In paragraph 12 the Examiner disagrees with the Applicant's previously submitted arguments that Rastogi et al. '449 does not have a logging mechanism capable of saving state at the sub-operation level, that is, at the step level, wherein a sub-operation, or a step, is an intermediate step executed in executing an operation.

In response, the Applicant refers the discussions herein above pertaining to, for example, paragraphs 7, 8 and 9 of the present Final Action.

The Examiner also states that the limitation of state logging for sub-operations does not appear in the claims.

In response, the Applicant refers the Examiner to the discussions herein above pertaining to paragraph 7, 8 and 9 of the present Final Action.

The Examiner also states that the limitations of the currently pending claims do not include the limitation, discussed in the previously submitted arguments distinguishing the present invention over Rastogi et al. '449, of continuous logging of state information at the at the sub-operation level, that is, at the step level. The Examiner maintains that the lack of this limitation allows the claims to be interpreted as stating that the logging operation occurs only once rather than for each sub-operation, or step, or an operation.

The Applicant believes that the Examiner is expressing what is essentially as 35 U.S.C. 112 issue as the Examiner seems of the opinion that the relevant language of the claims can be interpreted in more than one way.

The Applicant concurs that the claims do not include this limitation as an explicit recitation, and has amended the claims accordingly to remedy this lack. It will be noted that this amendment does not add any new matter to the specification or claims and does not alter the scope, meaning or subject matter of the claims, but has merely clarified the pertinent recitations.

Lastly, the Examiner has once again referred to the roll-back of active transactions as described in column 10, lines 52-61, although the Applicant is not sure of the point the Examiner is making by the reference at this point to the roll-back of active transactions. In response, however, the Applicant respectfully refers the Examiner to the discussions herein above of paragraphs 8, 9 and 11 of the present Final Action.

Paragraph 13:

In paragraph 13 the Examiner disagrees with the Applicant's position that Rastogi et al. '449 does not teach any form of state machine representing control and data values in a state machine and again refers to column 8, lines 24-25 of Rastogi et al. '449.

In reply, the Applicant refers the Examiner to the Applicant's responses herein above to paragraphs 7, 8 and 11 of the present Final Action.

Paragraph 14:

The Examiner addresses a number of issues in paragraph 14 of the Final Action, most of which are repetitive with the issues raised by the Examiner in preceding paragraphs of the Final Action.

For example, the Examiner disagrees with the Applicant's position and statements that Rastogi et al. '449 does not, at any point, describe a state machine or the saving of state, and objects to the statement as being repetitive with the arguments previously advanced by the Applicant. The Applicant respectfully wishes to point out that the argument objected to by the

Examiner is repeated from an earlier Response because the Examiner had repeated, from an earlier action, the same grounds for rejection that the argument answered.

Also, the Examiner had objected to the Applicant's arguments on the grounds that the Applicant had not cited the specific column and line numbers at which Rastogi et al. '449 teaches or fails to teach the use and saving of state and the operations of state machines referred to in the Applicant's argument.

In response, the Applicant refers the Examiner to the Applicant's replies herein above to paragraphs 7, 8 and 9 of the present Final Action, which cites column and line numbers in Rastogi et al. '449.

The Applicant also refers the Examiner to the Applicant's earlier discussion pertaining to paragraph 8 of the present Final Action, wherein the Applicant points out that Rastogi et al. '449 does not even mention "state" or "state machine" in the meaning used in the present invention at any place in the specification, claims or drawings, so that Rastogi et al. '449 is clearly not storing system state or state machines in any form as Rastogi et al. '449 does not even refer to the concepts of operation state or state machines.

As discussed with respect to paragraph 8 of the present Final Action, the only mention that Rastogi et al. '449 makes of the term "state" is when Rastogi et al. '449 is at column column 3, lines 22-35 and at column 5, lines 9-65 wherein Rastogi et al. '449 defines the term "state" as bits of information stored in the primary and secondary systems that indicate, in each system, whether the system is the current primary system or is the current secondary system and whether the two computer systems are in "synchronization", that is, have matching copies of the primary computer system transaction log.

The Rastogi et al. '449 definition of the term "state" clearly has no relationship to the use of the terms "state" and "state machine" as defined in the present invention. This clearly indicates that Rastogi et al. '449 did not have concept of "state' or "state machine" as used in the present invention, and thereby could not and did not describe any type of system using "state" or "state machine" in the sense of the present invention.

The Applicant has therefore pointed out and clearly identified the specific columns and lines wherein the Applicant believes that Rastogi et al. '449 supports the Applicant's position.

In passing, however, the Applicant does not understand, or has misunderstood, the Examiner's statement that the Applicant has failed to point out the specific column and lines at which Rastogi et al. '449 fails to teach the subject matter in question as it is very difficult to identify where something that has not been described is not described. The Applicant assumes that this paragraph was just unfortunately worded.

Lastly, the Examiner appears to quote a part of an argument previously advanced by the Applicant in support of the Examiner's position that the Applicant has acknowledged that the Rastogi et al. '449 sysem discloses the use of control and data values.

First, it must the noted that the quote from the Applicant's argument is taken out of context. The argument that is the source of the quote was advanced by the Applicant in support of the Applicant's position that although the Rastogi et al. '449 system does disclose control and data values, the Rastogi et al. '449 does not teach or suggest the use of state machines or the logging of state as in the present invention. The "acknowledgment", if the statement were to be taken in that sense, with which the Applicant does not agree, was that the Rastogi et al. '449 system does save and log control and data values as this is a necessary function in any logging system.

This is not in any way an acknowledgment or admission that the Rastogi et al. '449 system employs state and state machines in the manner of the present invention, but only that the Rastogi et al. '449 saves coarsely grained information regarding each operation performed by the Rastogi et al. '449 system. As discussed in the original argument, and as discussed herein above with regard to other paragraphs of the Final Action, it is the Applicant's position that the control and data values logged by the Rastogi et al. '449 are no more than the original instruction or command and input data initiating a given operation, not detail operation state information extracted and logged at each step of the execution of the operation.

Paragraph 15:

In paragraph 15 the Examiner once again disagrees with the Applicant's stated position that Rastogi et al. '449 does not teach or suggest anything beyond logging only the current instruction or command and input data, and that as a consequence Rastogi et al. '449 does not and cannot teach or suggest the logging of state or state machines, particularly at the sup-operation, or step, level. The Examiner again refers to column 8, lines 24-44 of Rastogi et al. '449 and the description of the Rastogi et al. '449 undo and redo log mechanisms.

In response, the Applicant refers the Examiner to the Applicant's above responses to paragraphs 7, 8, 9 and 11 of the present Final Action.

Paragraph 16:

In paragraph 16 of the Final Action, the Examiner expresses disagreement with the Applicant's position that Rastogi et al. '449 does no and cannot capture or record the sub-operations within a transaction. The Examiner again refers to the discussions in Rastogi et al. '449 pertaining to the Rastogi et al. '449 redo and undo logs as teaching the capture and recording of sub-operations.

The Examiner further states that the Examiner believes the Applicant's stated position to be merely an assumption made by the Applicant and asks the Applicant to point out specifically wherein in Rastogi et al. '449 the Applicant's position is supported.

In response, the Applicant refers the Examiner to the Applicant's responses to paragraphs 8, 9 and 11 herein above wherein the Applicant refers to specific sections of the Rastogi et al. '449 specification and explains in detail the Applicant's reasoning and conclusion to support the Applicant's position.

As also discussed herein above, it is the position of the Applicant that Rastogi et al. '449 does not in fact provide sufficient description of either the redo log or the undo log to comprise a teaching with regard to the present invention, or even of the Rastogi et al. '449 system, and is vague and inadequate even with regard to the Rastogi et al. '449 system.

In this regard, and for these reasons, the Applicant is of the impression that the Examiner is reading the teachings of the present invention onto and into the teachings of Rastogi et al. '449 to reach the conclusions expressed by the Examiner.

It is therefore the position of the Applicant that the Rastogi et al. '449 reference does not, in fact, comprise an adequate teaching under the requirements and provisions of either of 35 U.S.C. 102 or 35 U.S.C. 103.

It is further the belief and position of the Applicant, however, that in so far as Rastogi et al. '449 does contain a description of a redo and undo log and mechanism, that that description supports the conclusion and position of the Applicant that the teachings of pertaining the Rastogi et al. '449 redo log and undo are not pertinent to and do not disclose or even suggest the present invention under either 35 U.S.C. 102 or 35 U.S.C. 103. For example, and again as discussed herein above, the teachings of Rastogi et al. '449 pertaining to an undo log are not pertinent to the present invention because the Rastogi et al. '449 undo log and recovery mechanism perform an entirely different function than does the log mechanism of the present invention.

Paragraph 17:

In paragraph 17 of the Final Action the Examiner disagrees with a statement by the Applicant in the Response to the previous action to the effect that "It is noted, in this regard, that Rastogi et al. '449 does not in fact state that a log entry contains information taken during the execution of the transaction, but only that the stored information includes only the information essential to reconstruction of the transaction as an entity." The Examiner holds that the information taken during execution of a transaction is the same as the information essential to reconstruction of the transaction and states that it is unclear how the information differs.

The Applicant believes that the Examiner has mis-read or mis-understood the statement is question and apologizes for not making the meaning of the statement clear.

The essential meaning of this statement is that the type of information required to reconstruct a transaction, that is, to redo a transaction, depends upon the level at which the

transaction is to be redone. If a transaction is to be redone, or re-executed, on a step-by-step basis, that is, by redoing each individual, specific sub-operation in the transaction, then it is usually necessary to record the pertinent information existing at the start of each step.

It is very possible, however, to redo a transaction as an entity, that is, as a self-contained operation wherein the transaction is treated as a self-contained entity rather than as a sequence of individuals steps, or sub-operations, and it should noted that the statement in question refers to redoing "the transaction as an entity". This is the method typically used in older systems of the prior art, such as Rastogi et al. '449, because it requires the recordation of much less information than does the step-by-step method. That is, the redoing of a transaction as an entity typically requires the recordation of only the initial command or instruction stating the transaction to be performed and the initial input data to the transaction. This information is all that was required for a system to "do" the transaction the first time, and is all that a system typically requires to redo, or re-execute, a transaction as the system will thereafter follow its programming to execute, or redo, the transaction, just as in the initial execution of the transaction.

It should also be noted, for the sake of completeness, that as discussed herein above with regard to paragraphs 8, 9 and 11 of the Final Action, the "redo" of a transaction is a different type of operation than the "undo" of a transaction and may therefore require a different level of information than does the "undo" of a transaction. As also discussed herein above, however, the teachings of Rastogi et al. '449 pertaining to an undo log are not pertinent to the present invention because the Rastogi et al. '449 undo log and recovery mechanism perform an entirely different function than does the log mechanism of the present invention. For example, the Rastogi et al. '449 undo log and mechanism are a short term log that stores information pertaining to a transaction only during the actual execution of the transaction and discards the information as soon as the transaction is completed, which is in complete contrast to both the present invention and even the redo log and mechanism of the Rastogi et al. '449 system itself.

The Applicant hopes that the above discussion clarifies the meaning of the statement in issue, and invites the Examiner to discuss the matter further if the Examiner feels necessary.

Paragraph 18:

In paragraph 18 of the Final Action, the Examiner disagrees with the Applicant's statement that Rastogi et al. '449 does not even mention restoring or resuming a transaction at any of the sub-operations, or steps, comprising the transaction but only redoing or undoing as transaction, and again refers to column 10, lines 52-61 of Rastogi et al. '449.

In response, the Applicant refers the Examiner to the above responses to paragraphs 8, 9 and 11 of the Final Action, wherein the Applicant discusses and explains in detail the difference between the Rastogi et al. '449 undo and redo logs and mechanisms and the transaction information recorded therein and the logs of the present invention wherein information is recorded for each sub-operation, that is, state and state machine, of a transaction.

In addition, the Examiner also states that this argument is irrelevant because the limitation does not appear anywhere in the claims.

First, the Applicant cannot ascertain from the Examiner's statement whether the Examiner is saying that the limitations of restoring or resuming a transaction is not recited in the claims or that the limitation of recording information for each sub-operation or step of a transaction is not recited in the claims. The Examiner is incorrect in either case, however.

First, the Applicant would like to point out that each of the claims contains recitations directed to the extraction, or capture, and recording, or storing, of information for each sub-operation or step of a transaction wherein that information is represented by the system state for the sub-operation or step, which is in turn represented as a state machine for the sub-operation or step.

Secondly, the Applicant must also point out that each of the claims contains recitations reciting the restoration of transactions from the state machine information stored in the log and mirror logs and each of the claims contains recitations, either directly or by dependence, that

the state machine information, that is, the state information contained in each state machine, pertains to a corresponding sub-operation or step of a transaction. As a consequence, each claim contains recitations of the restoration or resumption of a sub-operation or step of a transaction.

For these reasons, therefore, it is the belief and position of the Applicant that these arguments are pertinent to the present invention as recited in the claims and distinguished the present invention as recited in the claims over the teachings and suggestions contained in Rastogi et al. '449.

Paragraph 19:

In paragraph 19 of the Final Action the Examiner questions the Applicant's argument regarding the two computers comprising the Rastogi et al. '449 system and rejects the Applicant's arguments regarding this issue.

In brief, the Applicant's argument is that in Rastogi et al. '449 the two computers are not parallel systems wherein both systems are concurrently engaged in the processing of transactions, that is, wherein each processes its own transactions. In the Rastogi et al. '449 system the two computers are separate and, at any given time, perform completely different functions. That is, only one system is actually processing transactions at any given time while the other system serves only to store the transaction log for the active system.

In a system of the present invention as recited, for example, in claims 3, 7, 11 and 15, the system is comprised of two sub-systems that normally operate concurrently and parallel, that is, with each executing its own transactions, which is a fundamentally different system from Rastogi et al. '449 wherein only one computer actually processes transactions at a time.

The Applicant has also reviewed Hemphill et al. '716, however, which is newly cited by the Examiner, and concurs that Hemphill et al. '716 does teach a system comprised of two sub-systems that normally operate concurrently and parallel, that is, with each executing its own transactions.

Therefore, and while Rastogi et al. '449 is not prior art with respect to this aspect of a system of the present invention, Hemphill et al. '716 does teach this aspect of a system of the present invention, although Hemphill et al. '716 does not appear to teach any other aspects of the present invention.

Paragraphs 20 and 21:

In paragraphs 20 and 21 of the Final Action the Examiner raises essentially the same issues that the Examiner raised in paragraph 19 of the present Final Action, and rejects the Applicant's previous arguments on this issue.

The Examiner also cites the Examiner's arguments addressing this issue in paragraphs 7 and 8 of the Final Office Action of January 12, 2004 in rejection of the Applicant's arguments on this issue.

In response, the Applicant refers the Examiner to the Applicant's above reply to paragraph 19 of the present Final Action.

Paragraphs 22, 23, 24 and 25:

Paragraphs 22, 23, 24 and 25 address essentially the same issues in progressively greater detail, that is, whether Kandasamy et al '314 teaches the capture and recording of data transfer requests as sub-operations or only as entities, and refers to those sections of the Applicant's previous Response wherein the Applicant discusses various related aspects of the Kandasamy et al '314 system in detail.

The Examiner respectfully disagrees with the Applicant's arguments, states that the Applicant's arguments are merely assumptions made by the Applicant, and states that the Examiner cannot find any support for the Applicant's arguments in the portions of Kandasamy et al '314 cited by the Applicant.

To consider the Examiner remarks in paragraphs 22, 23, 24 and 25 in greater detail, the Applicant refers the Examiner to column 3, lines 5-25, column 5, line 30 to column 6, line 20, column 7, line 22 through column 8, line 31,column 9, line 19 to column 11, line 56 of Kandasamy et al. '314. The following remarks are taken almost verbatim from Kandasamy et

al. '314, and the conclusions drawn from the descriptions provided by Kandasamy et al. '314 are not mere "assumptions" drawn by the Applicant but are obvious conclusions that will be well understood and accepted by those of ordinary skill in the arts

To begin with, the Applicant has previously explained that Kandasamy et al. '314 describes a fault tolerant mirroring file server system in which one or more clients and two or more file servers are interconnected through a local area network and network protocol layers using packet communications.

A request for a file transfer from a client to a file server is received by and executed, or replicated, by both file servers, but the initially addressed file server will normally respond to the request while the other file server will respond if the initially addressed server does not. The file servers thereby operate in parallel and concurrently to handle each read or write request, although only one will actually transfer the read or write data to or from the client while the other replicates the data transfer, so that the data stored in the two servers is identical and so that either can therefore reply to any read or write request.

There are therefore a number of aspects of the Kandasamy et al. '314 system that must be considered, and which the Applicant has attempted to explain in previous Responses.

First, it must also be noted that Kandasamy et al. '314 does not describe the system as extracting, capturing or storing any form of state or state machine information representing a data transfer, and in fact does not even mention state or state machines, but instead describes the system only as storing a copy of the data transfer itself, that is, a copy of the tranfer request and the data actually transferred. Unlike the present invention, therefore, the Kandasamy et al. '314 does not capture, extract or record state or state machine information.

In further distinction between the present invention and Kandasamy et al. '314, it must be noted that in the Kandasamy et al. '314 system the records stored by the two file servers are actual and literal copies of the data transfers themselves. That is, each record includes the read or write request, a copy of the data that is actually transferred, and, usually, some acknowledgment that the transfer was completed. Again, therefore, and in complete support

of the above "conclusion", the Kandasamy et al. '314 does not extract, capture or record state information or state machines in any form, but merely stores direct copies of each transfer request and the corresponding data that is transferred.

It addition, it mut be noted that each data transfer is therefore essentially treated as a complete and self-contained entity comprised of at least the request and the data transferred. This conclusion is not only supported by the direct description provided by Kandasamy et al. '314, but is also supported in that it is apparent that the elements of the Kandasamy et al. '314 system involved and active in a data transfer operation include not only the requesting client but both file servers, all of which are completely and concurrently involved in a transfer at the same time, so that each transfer must be dealt with as an entity.

A further aspect of the Kandasamy et al. '314 system that must be noted is that Kandasamy et al. '314 describes the communications between the clients and the two file servers as being through a packet type communications network. That is, and as is well known in the arts, a packet type communication system divides the information to be transferred between a client and a file server, that is, the transfer request, the data to be transferred, and any acknowledgment/coordination/synchronization information, into packets of fixed size and format. The packet communication system then sequentially transfers as many packets as are required to contain the the transfer request, the data to be transferred, and any acknowledgment/coordination/synchronization information.

While it is possible and common to interleave packet transfers between different parties on a communication network, it is normally necessary to execute a packet transfer between two parties, that is to sequentially transfer the number of packets necessary to contain the information being transferred between the parties, as a single operation to avoid confusion among the packets arriving at the receiving party. This again further supports the conclusion that each data transfer in the Kandasamy et al. '314 system is completed as a single entity before a next data transfer is initiated.

It is therefore again the belief and position of the Applicant that Kandasamy et al. '314 has no teachings or suggestions that are relevant to the present invention under either of 35 U.S.C. 102 or 35 U.S.C. 103.

Next considering certain of the Examiner's statements from paragraphs 22, 23, 24 and 25 of the present Final Action, in paragraph 22 the Examiner states that Kandasamy et al. '314 describes that the transactions between the systems are concurrent, so that sub-operations would be captured.

It has been defined in the present Application, and in the many Official Actions and Responses, that sub-operations, or steps, are the lower level operations that comprised a higher level operation and that are carried out in executing the higher level operation. Concurrent operation, however, refers to the case when to systems or sub-systems or elements of a system operate at the same time to perform the same or similar operations.

There is therefore no connection or relationship between performing operations "concurrently" and performing "sub-operations" as these are two entirely different matters. The Applicant therefore respectfully disagrees with the Examiner's conclusion that "concurrently" implies the recording of "sub-operations" and feels that this conclusion is unsupported by Kandasamy et al. '314. The Applicant cannot address the issue further, however, as the Examiner has not explained the reason behind this conclusion but has simply stated the conclusion.

In paragraph 23 the Examiner disagrees with the Applicant's arguments regarding the operation of the Kandasamy et al. '314 as a packet communication system and the Applicant's conclusion that this operation means that data requests and transfers are executed a self-contained operations, each including a transfer request and a data transfer.

In reply, the Applicant wishes to point out that Kandasamy et al. '314 clearly describes the system as employing packet communications and as executing data transfers between file servers and clients. The conclusions that the Applicant draws from these descriptions in Kandasamy et al. '314 are not unsupported, but are commonly known aspects of both data

transfers and packet communications systems, which the Applicant has once again attempted to explain above. Again, however, the Applicant cannot address this issue further as the Examiner has merely stated a disagreement with the Applicant's conclusions and has not expressed any reasoning to support the disagreement with the Applicant.

In paragraph 24 the Examiner has stated that the Examiner does not understand the relevance behind the Applicant's statement that the invention is distinguished from Kandasamy et al. '314 because the system of the present invention operates concurrently and in parallel at all times.

As discussed herein above, a system of the present invention as recited, for example, in claims 3, 7, 11 and 15, is comprised of two sub-systems that normally operate concurrently and parallel, that is, with each executing its own transactions at the same time the other is executing its own transactions.

In the Kandasamy et al. '314 system, however, and as also discussed above, each file server replicates, that is, copies, each data transfer operation between the other filer server and a client. In a given data transfer operation, therefore, the requesting client and both file servers are fully involved in the transfer operation at the same time, with one file server performing the data transfer and the other replicating the data transfer. As a consequence, the two file servers cannot operate on different data transfer operations at the same time and thereby cannot execute two different operations concurrently and in parallel, as can the system of the present invention.

It is therefore apparent that this distinction is substantive and significant.

Lastly, in paragraph 25 the Examiner refers to the concurrent operation of two sub-processors in the present invention as discussed just above with reference to paragraph 24 and a previously discussed with reference to paragraph 19.

The Examiner states that the Examiner will not address this issue because the limitation of two concurrently operating, parallel sub-processors does not appear anywhere in the claims.

In response, the Applicant wishes to point out that the present and operation of parallel, concurrently operating control/processing sub-systems is explicitly recited in, for example, claims 3, 7, 11 and 15, and thereby in their respective dependent claims, so that this argument is pertinent and material and, among other distinctions, distinguishes the present invention as claimed over the cited references.

Paragraph 26:

In paragraph 26 of the Final Action, the Examiner expresses disagreement with the Applicant's position that the combination of Rastogi et al. '449 and Kandasamy et al. '314 is invalid because, as stated by the Examiner, the combination of Kandasamy et al. '314 with Rastogi et al. '449 would be an obvious improvement over Rastogi et al. '449 alone. The Examiner, however, does not show or suggest how the Rastogi et al. '449 and Kandasamy et al. '314 references could be combined to provide a resulting system having the claimed advantages.

The Applicant therefore continues to respectfully disagree with the Examiner for the reasons stated in the previous Responses, which are incorporated herein by reference. Briefly, and as discussed in detail in previous Responses, the systems taught by Rastogi et al. '449 and Kandasamy et al. '314 are too different in structure and principle of operation to be a valid combination and, as also previously discussed in detail, any possible combination of Rastogi et al. '449 and Kandasamy et al. '314 would result in a system not only not having the perceived advantage, but one have numerous additional disadvantages.

Stated another way, when combining references it is not sufficient only to perceive some advantage from the combination, it is also necessary to show how to combine the references into a working system having the claimed advantages. The Examiner has failed to make this showing, while the Applicant has described in detail the results that may accrue from such a combination and the very real disadvantages resulting from such a combination.

In summary, therefore, and for the reasons discussed in detail in previous Responses, which are incorporated herein by reference, and for the reasons discussed herein above with

respect to the Examiner's remarks, it is the belief and position of the Applicant that the present invention as recited in the claims as amended herein above is fully and patentably distinguished over and from the teachings of Rastogi et al. '449, Kandasamy et al. '314 and Hemphill et al. '716 and all combination thereof under the requirements and provisions of both 35 U.S.C. 102 and 35 U.S.C. 103. The Applicant therefore respectfully requests that the Examiner reconsider and withdraw all rejections of the claims, and the allowance of the claims as presented herein above.

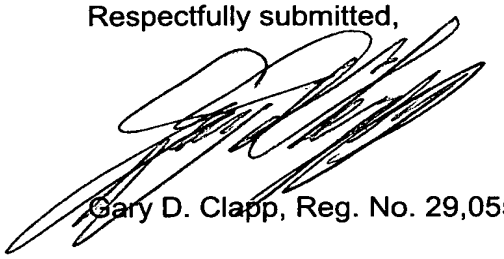
If any further amendment to this application is believed necessary to advance prosecution and place this case in allowable form, the Examiner is courteously solicited to contact the undersigned representative of the Applicant to discuss the same.

In view of the foregoing, it is respectfully submitted that the raised rejection(s) should be withdrawn and this application is now placed in a condition for allowance. Action to that end, in the form of an early Notice of Allowance, is courteously solicited by the Applicant at this time.

The Applicant respectfully requests that any outstanding objections or requirements, as to the form of this application, be held in abeyance until allowable subject matter is indicated for this case.

In the event that there are any fee deficiencies or additional fees are payable, please charge the same or credit any overpayment to our Deposit Account (Account No. 04-0213).

Respectfully submitted,



Gary D. Clapp, Reg. No. 29,055

Customer No. 020210
Davis & Bujold, P.L.L.C.
Fourth Floor
500 North Commercial Street
Manchester NH 03101-1151
Telephone 603-624-9220
Facsimile 603-624-9229
E-mail: patent@davisandbujold.com